PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 3/14	A1	(11) International Publication Number: WO 91/06050 (43) International Publication Date: 2 May 1991 (02.05.91)

(21) International Application Number: PCT/US90/06000

(22) International Filing Date: 16 October 1990 (16.10.90)

(30) Priority data:
423,785 17 October 1989 (17.10.89) US

(71) Applicant: APPLIED BIOSYSTEMS, INC. [US/US]; 850 Lincoln Centre Drive, Foster City, CA 94404 (US).

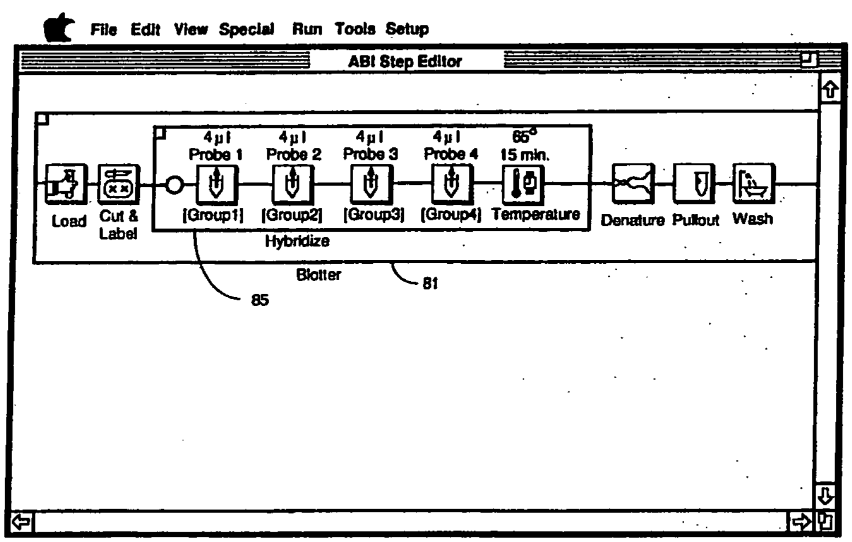
(72) Inventor: GUIREMAND, Harry, A. ; 454 Second Avenue, Half Moon Bay, CA 95019 (US).

(74) Agent: SMITH, Joseph, H.; Applied Biosystems, Inc., 850 Lincoln Centre Drive, Foster City, CA 94404 (US).

(81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).

Published
With international search report.
With amended claims.

(54) Title: ROBOTIC INTERFACE



(57) Abstract

An automated apparatus (11) is programmed to perform a process by arranging a sequence of first icons (69) on a display (15) in the order of the process, wherein the first icons (69) represent functions of the apparatus (11), and wherein at least one of the first icons (69) provides a visual representation of a function of the apparatus (11). Said at least one of the first icons can be expanded to show second icons (81) that comprise the function of said at least one of the first icons (69), and at least one of the second icons (81) provides a visual representation of a subfunction of the apparatus (11). In a preferred mode, when said at least one of the first icons (69) is expanded, said at least one of the first icons (69) maintains its same sequential relationship on the display (15) to the other of the first icons (69) in the sequence as before it was expanded.

BEST AVAILABLE COPY

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MR	Mauritania
BE	Belgium	GA	Gabon	MW	Malawi
BF	Burkina Faso	GB	United Kingdom	NL	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IT	Italy	RO	Romania
CA	Canada	JP	Japan	SD	Sudan
CF	Central African Republic	KP	Democratic People's Republic of Korea	SE	Sweden
CG	Congo	KR	Republic of Korea	SN	Senegal
CH	Switzerland	LI	Liechtenstein	SU	Soviet Union
CI	Côte d'Ivoire	LK	Sri Lanka	TD	Chad
CM	Cameroon	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark				

ROBOTIC INTERFACE

Field of the Invention

The present invention is in the area of user interfaces for computers, and has particular application to the programming of automated processing equipment.

Background of the Invention

Programming computers of all sorts has become a very sophisticated art, and computerized techniques for building other programs and for accomplishing such tasks as altering program flow and changing values of variables in a program are commonplace. One area of computer application where such techniques have found considerable application is in the area of controlling automated processing equipment.

It is often true in controlling automated processes that there are several steps in a process, and that one or more of the several steps might be done in a variety of ways, or under a variety of conditions. It is also true that a particular step might be performed more than once at different times in the overall process flow. For example, consider a machine for performing chemical procedures automatically that are ordinarily performed by hand. Such a machine might have a robotic system for moving chemicals to specific stations on the machine, where specific kinds of procedures might be performed, and storage areas where containers of various chemicals, solvents, reagents, and other supplies might be stored. One station could be used for heating substances in the containers to facilitate or precipitate chemical reactions. There could be another

station for stirring. Yet another station might be devoted to diluting the contents of a container by addition of a particular solvent or other chemical. There are many other procedures that could be dedicated to specific stations, or even performed in a single station.

In the performance of chemical procedures with a machine such as described above, it would be an advantage if the order of procedures could be quickly and easily altered. In one case for example, a user might wish to select a particular sample material from one position in a storage matrix of test tubes, and then move it through a sequence of process steps. He might wish to heat the sample, stir it, add an enzyme solution, stir again, heat again, add another chemical, allow time for a reaction, then replace the sample in the storage matrix at a different position from the original position, afterwards retrieving a different sample for an entirely different procedure.

A machine with the robotic ability to alter the sequence of steps as described has special requirements that are necessary for its flexibility in use. Not only must the sequence of steps be programmable, but the conditions under which each step is performed need to be programmable as well. For example, at a heating station, the researcher needs to be able to choose the temperature to which a sample is heated, and perhaps the length of time that the chosen temperature is maintained as well. At a stirring station an operator may want to control the vigor with which a mixture is stirred and the time duration for the stirring action. There are a number of variables at each station in such a machine that an operator might want to control.

The value of such a machine is determined in large part by its flexibility and the ease and accuracy with which changes in the order of procedures and the magnitude of process variables may be altered. To provide a computerized operating system for such a machine is no simple task. If a program is written in which the subroutines to perform the various steps are sequentially entered and there is no way to send the program flow from one to any other subroutine at a signal, then the ability to alter the sequence of steps is sacrificed. A program for such a machine must be modular in the sense that control subroutines must be callable and arrangeable in almost any order for program flow. Also within each control subroutine corresponding to a definable task, there needs to be a way for an operator to alter the variables for each type of activity, for example: temperature at a heating station.

In addition to modularity and facility in entering variable values, there needs to be an interface for the operator that allows the operator to choose steps in a sequence with reference to other steps and to maintain a sense of position in the program. Without the position sense, an operator might, for example, easily enter variable values for one step that were meant for another.

Although it is possible to enter steps and variable values for such a program by reading storage media that can be programmed off-line, such as floppy disks or recording tape, it is generally desirable to alter step sequence and enter variable values on-line through an operator interface. By providing on-line interaction, an operator may make corrections and adjustments quickly and easily. An operator interface, however, is needed whether the programming is done on-line or off-line.

Typically, an operator interface for such a control program is an interactive display on the screen of a monitor, where an operator may reorder the steps and enter values for variables in one or a combination of ways. One way is for the program to ask printed or audible questions and then go to an input mode for an operator to enter a sequence number, a choice from a list, or a value from a keyboard, which input the computer program will store in a particular memory location to be accessed at a later time during execution of the sequence being programmed.

Another interface mechanism is called menu-driven, in which the program presents lists of choices on the display, and the operator may select by operating a cursor with the aid of a device called a mouse, providing a signal by pressing a button on the mouse. In a menu-driven interface text fields may also be presented for entry of information from the keyboard.

In other kinds of interfaces, a process flow may be simulated on a display device by the program as graphic symbols for various processes with the symbols connected by lines representing continuity in the process flow. In some cases an operator can move symbols from one position to another in the flow chart and may enter information for variables at the position of a symbol representing a process.

A problem with presenting a process flow schematic on a CRT or other computer display is that the area for display is limited. If a process flow schematic has a relatively large number of nodes, a node being a symbol or other notation representing a specific activity, it may not be possible to present all of the nodes on the screen. Presenting all of the nodes becomes

particularly difficult if there are text fields involved or menus to allow entry of process variables at some or all of the nodes.

One manner in which the space problem has been addressed is by dedicating separate screens to separate parts of a program, with a portion of an overall process flow presented on each screen. In this implementation of a process flow schematic, there has to be a menu or other device for an operator to select a screen.

A node in a process flow schematic for machine control is typically a box, triangle, or other polygon symbol with identifying text. A node may also have one or more text fields for entering information. A text field is a field identified on the display that can be selected to become active, and will then accept keyboard input. The input is typically displayed in the text field by the computer program as well as stored at an assigned memory address for later reference. If nodes in such a program contain text fields, then the display space problem is more critical, fewer nodes may be displayed, and more screens are required for display of an entire process flow schematic.

The relationship of nodes to one another in a process flow is generally sequential, with alternative (parallel) paths sometimes encountered. The relationship of information at a node, however, is typically hierarchical, i.e. a conceptual, logical ordering of the information at each node, to condense content, and to more directly relate the primary concepts from one paragraph to those of another. One way of handling hierarchical information on a display to save space is typified by outline programs such as Thinktank, MaxThink, and More, marketed by Living

Videotext Co., Inc. of Mountain View, Ca. In an outline program, an outline may be collapsed so that only major headings are shown.

From a collapsed outline entered in a program like Thinktank, a user may, with an appropriate signal, e.g. positioning a cursor on a heading and pressing a pushbutton, cause entries at a next level to be displayed while the original level is still on the display. The next lower level is typically shown indented one position from a next higher level. Further expansion is accomplished in the same manner, and, as is typical with an outline, the lowest level may be paragraphs of text rather than a heading. One may then view the outline in various states of expansion and collapse, accessing those parts of the outline needed at any particular time.

A similar mechanism to the collapsing and expanding outline is provided by a windows ability which is common to programs for Apple Computers such as the Macintosh line. By "clicking" on a graphic symbol, a window is displayed. Window type programs, e.g. Microsoft Windows and X Windows, are also becoming increasingly popular for other computers such as IBM Personal Computer compatible systems and for larger systems such as systems using a UNIX operating environment.

"Clicking" is short terminology for positioning a cursor on the symbol and pressing a button. A window is a screen area, like a smaller display, positioned over the original display seemingly at another level, as another layer. A window seems to float on the original. Lower order windows in a hierarchy may be floated over other windows. In the Macintosh Finder system and many applications, windows may be moved from

one apparent level to another, and moved to different positions on a display by a process called dragging, which is similar to clicking. Microsoft Windows for IBM compatible machines is marketed by MicroSoft Corporation of Redmond, WA.

Windows may be programmed to have text fields, menus, check boxes for selection, and other interactive features. A serious disadvantage is that with windows the relationship of a node to other nodes in a process flow is lost when a window is opened. There is no visual link to a flow schematic when the window is opened.

There are, then, several distinct shortcomings in interactive operator interfaces for creating process schematics and other sequential programs, and for editing node-specific variables and other hierarchical information at a node. One, as mentioned earlier is an inability to display entire complicated schematics on a single screen. Another is a loss of relational information with expanding and collapsing techniques such as windows. Still another is a sameness in the look of nodes, requiring reference to numbers or written descriptions for identification, which may easily lead to error. Yet another is a typical requirement to collapse nodes before choosing different screens to see other parts of an extensive process flow schematic.

What is clearly needed is an interactive interface with nodes represented by descriptive graphic symbols so specific activities and activity strings may be easily recognized. Such an interface should retain relational information of one node to surrounding nodes in a process flow regardless of state of expansion or collapse hierarchically. Also, to avoid the problem of

calling other screens, the program should provide a display that is at all times a window on a full process schematic, allowing a user to pan to other parts of the schematic without changing screens. Lastly, for applications involving robotic process control, the interface should not be just a process flow visualization means, but should be keyed directly to the apparatus it is related to, so that the apparatus is operated via the interface.

Summary of the Invention

In accordance with preferred embodiments of the invention a powerful and flexible robotic control language is disclosed which frees the user from the characteristic dialects used in computer programming. The language provides an advanced user interface for programming the function of complex equipment, e.g. such as laboratory robots. The interface facilitates rapid learning as well as a convenient day-to-day tool for use by experts. This is achieved for robots through the use of an iconic programming language for the description of robot operations, and an intuitive interactive enviroment for creating, editing, and executing programs. The iconic programming language primarily differs from traditional text-based ones in that its symbols are icons rather than words, with the advantage that icons are immediately recongnizable, independent of natural language, cannot be misspelled, and a single symbol can represent many interconnected functions. The function of syntax, the formal relationships between symbols, is fulfilled in this visual programming language via ordered, and enforced, ways of placing symbols in relationship to one another.

According to a method of the invention, an automated apparatus is programmed to perform a process by arranging a sequence of first icons on a display in the order of the process, wherein the first icons represent functions of the apparatus, and wherein at least one of the first icons provides a visual representation of a function of the apparatus. Said at least one of the first icons can be expanded to show second icons that comprise the function of said at least one of the first

icons, and at least one of the second icons provides a visual representation of a subfunction of the apparatus. In a preferred mode, when said at least one of the first icons is expanded, said at least one of the first icons maintains its same sequential relationship on the display to the other of the first icons in the sequence as before it was expanded.

Brief Description of the Drawings

Fig. 1 is a perspective view of an automated laboratory for perfoming chemical processes.

Fig. 2 is a block diagram of a control interface for the automated laboratory of Fig. 1.

Fig. 3 shows a sample of exemplary icons that can be used with the automated laboratory for different laboratory functions.

Fig. 4 shows a screen of a Proto program according to the invention for a DNA labelling and hybridization process called Blotter.

Fig. 5 shows a display of a first level expansion of the icon representing the Blotter process.

Fig. 6 shows a display of a second level expansion of the Blotter icon, which illustrates the Hybridize subprocess.

Fig. 7 shows a display of a third level expansion of the Blotter icon, which illustrates the Temperature subprocess within the Hybridize subprocess.

Fig. 8 shows a display of a second level expansion of the Blotter icon, which illustrates the Cut and Label subprocess.

Fig. 9 shows a third level expansion of the Blotter icon, which illustrates the Add Restriction Enzymes subprocess of the Cut and Label subprocess.

Fig. 10 shows a fourth level expansion of the Blotter icon, which illustrates the Group 4: Hodgkins Disease subprocess within the Add Restriction Enzymes subprocess.

Fig. 11 shows a fifth level expansion of the Blotter icon, which illustrates the Ligaid Q+ subprocess.

Fig. 12 shows a Files menu.

Fig. 13 shows an Edit menu.

Fig. 14 shows a View menu.

Fig. 15 shows a Special menu.

Fig. 16 shows a Run menu.

Fig. 17 shows a Tools menu.

Fig. 18 shows a Setup menu.

Fig. 19 shows a window for selecting Group samples.

Fig. 20 shows a shows a window for selecting Restriction Groups.

Fig. 21 shows a window for selecting Probes.

Fig. 22 shows a diagram of the code structure for the Proto programming language.

Description of the Preferred Embodiments

This invention is related to an interactive user interface for programming process equipment and handling and storing other relational and hierarchical information. Hence, its description will be put in the context of a specific application to illustrate its use and power. Those skilled in the art will appreciate that the context is used merely for illustration and that other examples of automated equipment could also be used.

Fig. 1 is a perspective view of an automated laboratory (AL) 11 for performing chemical processes, e.g. such as those involved in molecular biology. A computer 13 with a CRT monitor 15, a keyboard 17 and a mouse device 19 is connected to the AL. The computer, CRT display, mouse, and keyboard are hardware components of an operator interface for programming the AL to perform sequences of activities, for starting and stopping processes and sequences of processes and for entering and altering process variables for specific activities. In the preferred embodiment the computer is a Macintosh II computer made by Apple Computer of Cupertino, CA, but other computers may also be used.

The AL has a closed heating station 21, a cold storage station 23, a wash station 25, a storage position 27 for storing and presenting frequently used fluids such a DI water and solvents, and a separation station 29 for separating materials in suspension in sample fluids. A transport apparatus 31 carries a needle 33 of a pipette system for aspirating fluids from containers at the various stations and dispensing the same fluids into containers at the same or other

stations. The pipette system includes two syringe pumps (not shown) in the preferred embodiment. One pump is for relatively course transfer, and the other is transfer of precise amounts of fluids. There are also actuators, motors, sensors, printed circuit boards, power supplies, and other devices (not shown) typical of such equipment. One end of the AL at region 46 is shown cut away so internal details may be seen.

Heating station 21 has positions for placing a plurality of vials for holding samples and mixtures of fluids, and lid 45 is automatically closed to seal the tops of all the vials in the station during heating. Cold storage station 23 has an array of vial positions in the preferred embodiment similar to the array at station 21, except the lid thereon does not open. The lid has holes through which the needle of the pipette may pass to aspirate or dispense fluids. Storage position 27 is an indented area where a user may place carriers holding a number of tubes of frequently used fluids. Separation station 29 has two rows of positions for vials for sepating materials that are suspended in solutions by performing an extraction process. The separation station also has a resistance heater for maintaining heat of fluids at the station and water cooled passages, also for temperature control.

Transport apparatus 31 moves along slot 35 passing over the storage and activity stations. The pipette needle is movable along arm 37 of the transport device in the direction of arrow 39 and the transport is movable along slot 35 in the direction of arrow 41 to position the pipette over any container position at an activity station. The pipette needle is translatable vertically as well in the direction of arrow 43 so the

transport apparatus is an XYZ mechanism capable of placing the pipette in any container on the AL.

The pipette is for aspirating fluid from any one container and dispensing it into any another container. With the pipette, mixtures of various chemicals and other fluids may be made, diluted, and concentrated, and transported to any vial or other container on the AL. The pipette system may also serve to agitate fluids in a container to perform mixing, by repeated aspirating and dispensing of the fluid in a container. Wash station 25 is used for washing and backflowing the pipette tip after a transfer of fluid to avoid cross-contamination.

Computer 13, CRT 15, mouse 19 and keyboard 17 are used with a unique program, hereinafter called Proto, to prepare control sequences and establish specific characteristics for the various activities that make up a complete control sequence, as well as to initiate and terminate specific strings of activities. Fig. 2 is a block diagram showing control activities and modules.

Computer 13, keyboard 17, mouse 16, and display 15 are connected together in the usual way, and the computer is connected by a communication line 47 to an electronic interface module 49. A variety of communications protocols can be used. However, since the system is implemented using the Apple Macintosh, the Appletalk protocol is preferred. As is customary in the art, module 49 includes switching elements for converting low-level signals to voltages and currents capable of running the actuators and motors of the AL, power supplies, analog-to-digital converters and digital-to-analog converters, among other equipment needed to translate low-level signals from the computer to levels sufficient to drive the activities of the AL.

Line 51 represents multiple circuitry to carry both communication and power between module 49 and specific drives, actuators and sensors on the AL.

X-Drive 53, Y-Drive 55 and Z-Drive 57 are the three dimension drives for transport apparatus, and each of these drives also includes position sensors that send signals back to the computer via module 49 so the computer has updated information relative to the position of the transport elements. Aspirate actuator 59 causes the travelling pipette to aspirate fluid from a container after the pipette is positioned, and dispense actuator 61 causes fluid to be dispensed from the pipette. In the case of both aspirate and dispense there are timers and sensors (not shown) that control the rate and amount of fluid aspirated or dispensed.

Heater control 63 is for heat at the heater station, and includes thermal sensing elements to inform the computer of the effect of heating. Separation control 64 manipulates assemblies at the separation station. Refrigeration control 65 is for maintaining temperature at cold-storage station 23, and includes temperature sensing. Wash actuator 67 operates the washing action at the wash station to wash the pipette between transfer procedures that use the pipette. Mixing actions in the preferred embodiment are accomplished in vials and other containers, if needed, by repeated aspiration and dispensing of fluid.

As indicated earlier, a unique program, Proto, is run on the computer in the preferred embodiment to create control programs, enter and edit variable values, and initiate and terminate process sequences. Proto is an iconic program, employing graphic symbols called icons to represent processes, process steps, and other

activities. Proto provides a unique user interface that is useful for handling hierarchical information and for controlling many kinds of process machines and equipment.

Proto has a set of routines allowing a user to select icons representing various activities and to organize the icons into flow schematics representing process flow, with the icons connected on the display with lines. The icons may also be nested such that a relatively complex sequence of activities may be represented by a single icon, and the single icon may be expanded in place to show a connected sequence of icons representing steps in the more complex sequence. The second level icons may also consist of sequences of other icons, also expandable in place, until, at the lowest level, icon sequences consist of icons representing fundamental process steps. The fundamental steps in the preferred embodiment are typically themselves sequences of even more basic activities. For example, a fundamental step may be a direction by the program to the AL to send the transport to the Home position. The control system through sensors tracks the real-time position of the transport apparatus and the Home position is a known position to the computer, so the computer would then operate the X-drive and the Y-drive in the appropriate directions to attain the home position. Fundamental steps, however, could be as basic as to move the z-drive in one direction by a specific amount.

Fig. 3 is a sampler showing a number of the icons provided with the Proto program for use in building and operating control programs to operate the AL. The icons shown are labeled as to their function, and only a

selection of the implemented icons is shown. Proto provides an ability for a user to define new icons and to specify the relationship to function of new icons in a program; that is, what code is to be inserted into a program and what signal string is to be generated and sent when an icon is activated.

A fundamental icon is typically expandable in the preferred embodiment to display an input window which allows an operator to enter or edit variable values for the fundamental process step represented by an icon. In addition, the graphic icons are drawn to uniquely represent the steps, sequences of steps, or complete processes that they denote. The icons are verbs in this sense. As an example, an icon for Disepense is a small picture of a test tube showing the direction of flow of materials into the tube. This icon can then be expanded to illustrate the fundamental processes that are used in the dispense function.

In programs other than Proto, such as programs prepared in Apple Computer's Hypercard application or in Microsoft's Windows program, expanding a symbol results in a window floating over the icon, and the network representation, if any, appears as though the window is in a separate graphic layer on the screen. Indeed, such windows may often be repositionable on the screen by an operator and caused to show in a different layer, in front of or behind other such windows. Hence, the relationship of the symbol to the rest of the elements on the screen is lost.

In Proto, the icons are expandable in place, and the new network or control panel displayed by expansion of an icon is shown still connected to the network sequence on both sides. Moreover, the previous icon, though no

longer visible, is still represented as a border of a box, surrounding the lower order network, relaying the continuing relationship to the user. The expansion in place allows a user to maintain an accurate sense of place in a network of icons, and reduces the probability of costly error in placing and removing icons to alter a process sequence, as well as in entering and editing variables for specific activities. Moreover, even though expansion creates greater competition for screen space, as in other programs, Proto does not resort to separate screens, breaking the relational continuity, but provides for scrolling (panning) of a single screen in the expanded state so a user may view the entire process network in any expanded or collapsed state.

Fig. 4 is a screen of an actual program, called Blotter, prepared in Proto for controlling the AL to perform a process for labeling selected DNA sequences. In the screen of Fig. 4, the process network is in its most collapsed state, reduced to a single blank icon 69 named Blotter. A menu bar 71 contains labels that are selectable to activate programming functions for creating and editing programs in Proto. Blotter icon 69 is the icon for the fully collapsed Blotter control program to operate the AL or other processing equipment. There is a vertical scroll bar 75 and a horizontal scroll bar 77 for scrolling the screen. In the fully collapsed state, as in Fig. 4 the scroll bars are shown blank because the screen does not need to scroll. Single icon 69 represents the entire control program.

A cursor 79, showing on the display as an arrow at a slight angle, is movable throughout the display area in response to movement of mouse 19 (Fig. 1). Cursor

movement may be accomplished by other devices as well, such as a joy stick, a palm ball, a puck, and by keys on the keyboard. Selection on the display is by positioning the cursor at a particular point and momentarily pressing a button on the mouse. This combined action, as indicated earlier, is termed "clicking" in the art, such as "click on the Blotter icon". This terminology will be used hereinafter in the present specification. A different action is initiated in some instances by "double clicking", which is positioning the cursor and pressing the button twice in rapid succession.

By clicking on the Blotter icon, a user may select the Blotter process for some action or editing. Typically, in Proto, the item clicked is then shown highlighted in the display, and any menu function chosen and activated will act upon the highlighted process. Double clicking in Proto will expand an icon in place to reveal subprocesses arranged in an order that make up the process represented by the icon, or in the case of an icon at the fundamental level, will display a variable entry box, called the control panel for the particular icon, for making choices and entering variables to alter the characteristics of an action initiated by the icon.

A first level expansion is shown in Fig. 5, which is the display after double clicking the Blotter icon. In Fig. 5 rectangle 81 represents the Blotter process that was represented in Fig. 4 by icon 69. Within rectangle 81 there is a sequence of seven subprocesses: Load, Cut and Label, Hybridize, Denature, Pullout, Wash, and Remove; connected by lines, indicating clearly to a user that the Blotter process is made up of these seven

subprocesses in the order shown with the convention from left to right for time sequence. The relationship of one process to another and the hierarchical relationship are clearly apparent.

Rectangle 81 representing the Blotter process has a small rectangle 83 in the upper left corner that is known as the close box. Clicking on the close box collapses the representation of the network shown in Fig. 5 back to the single icon as shown in Fig. 4. Every expansion box has a close box for expedient closing to collapse to a higher order display. Typically collapsing, i.e. closing, may also be accomplished by highlighting the box and choosing a close command from the file menu in the menu bar, or by highlighting and a keystroke combination.

A user may click on any one of the seven subprocess icons of Fig. 5 to highlight that icon for editing or other action, or double click on any one to open that subprocess to a next lower level of detail. Clicking once inside rectangle 81, but with the cursor not on any one of the seven subprocess icons will highlight the Blotter rectangle, which is functionally equivalent to highlighting the Blotter icon in Fig. 4.

Fig. 6 shows a new display, the result of double clicking the Hybridize subprocess to open it. This is a second level expansion. Now there are two rectangles on the screen, rectangle 85 representing the Hybridize subprocess and rectangle 81 representing the Blotter process. Rectangle 85 is shown nested within rectangle 81 along with the other six subprocesses that make up the Blotter process. The relationship of the Hybridize subprocess to the other six subprocesses is clearly represented by connecting lines, and now the Hybridize

subprocess has been expanded in place to reveal a sequence of five lower order activities that make up the hybridize subprocess: Probe 1, Probe 2, Probe 3, Probe 4, and Temperature.

In the expanded state of Fig. 6 not all the network may be seen on the monitor screen, because the two expansions have taken more than the full horizontal dimension of the screen. The complete network in its expanded state is still apparent on the single display, however, because the display can be panned to the left and right by use of the horizontal scroll bar. The display may also be panned by a process called dragging. The user can position the cursor at a point outside any icon and press and hold the mouse button while moving the mouse. The image on the screen will move as though a document bigger than the screen is being moved behind the screen. The screen operates as a movable window on a document larger than the screen.

Although it is not shown in Fig. 6, other icons could be expanded without closing the Hybridize icon, and the others would also expand in place without losing any sense of relationship to the rest of the network. By panning, the entire network could still be moved into view.

Fig. 7 shows the result of double clicking the Temperature icon of Fig. 6, which displays control panel 87 for allowing a user to input and edit characteristics for temperature control. The Temperature control panel in this case has text fields for Rack, Temperature, Ramp Hold, and Failsafe range. For the entries as shown, when the Temperature icon is activated in the process flow, the temperature of a rack known as the Reaction rack will be cycled to a temperature of sixty-five

degrees centigrade at a maximum ramp (as quickly as possible), and the temperature will be held for fifteen minutes. There is also a text box to enter a failsafe range, which is none in the instant case. A control panel has a close box like a process rectangle, and a user may close the temperature control panel after making or editing entries, collapsing the display back to the display of Fig. 6.

It is not required that each subprocess have the same number of nested levels. Fig. 8 shows the network with the Hybridize subprocess collapsed to the Hybridize icon, and the Cut and Label subprocess expanded to the next level. Rectangle 89 represents the Cut and Label subprocess, which in Fig. 7 is seen to consist of two other subprocesses: Add Restriction Enzyme Groups and Label. Adding restriction enzymes is a part of the chemical procedure by which DNA molecules are cut.

Fig. 9 shows the display after double clicking on the Add Restriction Enzymes subprocess, which is represented in Fig. 9 by rectangle 91. The Add Restriction Enzyme subprocess is revealed to be a choice of one of four parallel paths, depending upon which restriction enzyme group a user elects for a particular Blotter process. This permits changes in the chemistry to be made for any particular path.

Fig. 10 shows a further expansion as the result of double clicking on the icon of Group 4, Hodgkins Disease, in the Add Restriction Enzyme Groups rectangle. Rectangle 93 in Fig. 10 represents the Hodgkins Disease icon of Fig. 9. Fig. 10 shows that there is a sequence of 5 restriction enzymes to be added for the Hodgkins Disease group. Fig. 11 shows the result of double clicking on the Ligaid Q+ icon in Fig.

10. Control panel 95 is for setting control variables for aspiration of Ligaid Q+ by the pipette in the Blotter process. Fig. 11 shows text fields for setting the volume, choosing the source, adjusting the pump rate, choosing the level at the source at which to aspirate, setting the ascend and descend speeds for the pipette drive, and setting an airgap and a predispense volume.

To completely define a process, a user opens the necessary icons, expanding to the level needed, enters the necessary values to define the process, and then collapses the network again. A Blotter program thus defined can be saved to be used as needed to control the same process.

Proto includes tools, accessible in menu bar 71, for altering the sequence of icons, entering new icons, deleting icons, changing names, and doing all functions necessary to build programs. By clicking on a choice in the menu bar an operator causes a menu to appear. The operator may then drag the cursor down the menu to a listed function, and the function will be activated when the operator releases the mouse button. This operation is a well known procedure in menu-driven programs.

Some of the menus in Proto are common to many Apple programs, and will be recognizable to a programmer with skill in the art. For example, choosing the Apple logo in the menu bar displays a menu of functions that are loaded as desk accessories, such as Chooser, Scrapbook and others.

The Files menu, shown in Fig. 12, is for activating functions that relate to such activities as opening, closing, saving and printing files in Proto. In many cases a function may be activated from the keyboard by a

key combination without clicking on the choice in the menu bar. If so, the shortcut key combination is shown next to the function label in the function list. In the Files menu, choosing New presents a new window on the screen which may be used to create a new control program by the use of other functions from the menus, described in further detail below. Open presents a scrolling window by which an operator may access each storage device, usually disk drives, that are operating for the computer, may list the Proto programs on each storage medium, and may select a program for display and editing. In the preferred embodiment, the scrolling window presented and the listings are similar to those familiar to Macintosh users.

Close removes the display of a Proto program from the screen. Save causes an edited program being displayed to be saved to a storage medium, where it will be available to be loaded (opened) at a later time. Save as... allows an operator to save an open program to a storage medium under a different name, and provides a dialog window for the operator to enter the new name. The Save as... function makes it possible for a user to start with an existing program to edit when creating a new program that will be similar to the existing program. Save a Copy as... is a similar function to Save as... except the old program is left displayed on the screen after using the function rather than the new.

Revert in the Files menu causes the Proto program displayed to revert to the form that was last saved. Page Setup is a function familiar to Mac users that allows the user to specify for the Mac what sort of page layout will be used for printing; such as A size, legal size, C size, etc. Print is the command function to

send the current file to the printer specified in the Chooser function for hardcopy output. Quit is a function that closes the current file and closes the Proto program to the Macintosh desktop in the preferred embodiment. All of the File functions are functions familiar to Macintosh users.

The Edit choice from the menu bar presents a menu list of functions that are useful for editing Proto programs. The Edit menu is shown in Fig. 13. These functions are also familiar to Macintosh users, as they are used in other programs written for the Macintosh Copy, for example, used with a word processing program, will typically copy a selection of text into a memory location reserved for that purpose, where it may be accessed to be pasted into an open document at a different position. In graphics programs Copy is used to copy a selected region of pixels, or a selected vector or group of vectors, which may then be accessed to be pasted back into a picture at a different location, without removing or altering the original selection. Copy in Proto operates differently. In Proto Copy copies not only a selected icon, but all nested icons at a lower level in the network order that hold at the time of the copy, all control panels associated with icons, and, critically, the programming code associated with each of the icons and control panels. A Paste function after a Copy, then, will paste into a process network the copied icon, the lower order elements, and all the associated program code.

Undo from the Edit menu is a function that cancels the result of the last function performed. It is used typically to correct the results of a mistake. Cut removes a highlighted icon, all lower order icons

associated with the highlighted icon, all control panels associated with lower order icons, and all the code associated with the icons and control panels. The network is rejoined where the highlighted icon is removed. With Cut a copy of the elements removed is stored in the memory location reserved for Paste, so the elements removed may be reinserted at another point in the program network.

Copy was described above, as was Paste. Paste may be used to insert a copy of elements as a result of either a Cut or a Copy function. Clear removes a highlighted icon, lower order icons and control panels associated with the highlighted icon, and code associated with the other elements, without saving a copy for Paste. Clear is a functional eraser. Set Color allows a user to control the color of elements displayed in the Proto program, such as the background color of a process rectangle or an icon.

The View menu shown in Fig. 14 offers functions that control the way that a user may view displays of a Proto program. Zoom In allows a user to magnify a display and Zoom Out allows a user to return a display to normal, or smaller, from a magnified view. View as Text works by displaying elements in the display as text rather than as graphic symbols. Hide Notes allows a user to display a program without text notes on the display.

Special has a two-item menu list: Check Method and Recalculate Volume. The Special menu is shown in Fig. 15.

Check Method checks that the entire network of icons is internally consistent, and executable within the limitation of the designated hardware. Recalculate

volumes calculates volumes of fluids needed to perform the method and updates the load steps within those volume amounts.

The Run menu shown in Fig. 16 provides menu functions to control the starting and stopping of process flow performance according to a Proto process network. Start causes a process sequence to begin at the first step, which then proceeds through the steps represented by the icons in the Proto network. Suspend causes a process sequence to halt at the time and at the position in the process flow when and where a Suspend function is activated by a user. Resume causes a suspended process flow to resume. Abort causes the process flow to cease, and control to return to the beginning of the process flow. After Abort, Resume has no effect.

The Tools menu shown in Fig. 17 provides icons representing system functions. By highlighting an icon in a process network displayed in a Proto program, then selecting an icon from Tools, the icon, any nested functionality and control panels, and associated code for the elements represented, are inserted into the process flow network at a position immediately following the highlighted icon. The Tools function allows a user to build programs beginning with a blank screen with no icons to copy.

The icons of the Tools menu are the "lowest-level" icons, which represent the last level of expansion before a control panel. For reasons of complexity, these icons are provided in the program by coding in the computer language used to implement the Proto program. In the preferred embodiment, the language is the well known C-language.

Other aspects of the system can be implemented in the iconic language as well. For example, a Setup menu shown in Fig. 18 can be used to provide an ability to easily specify using the keyboard or mouse certain arrangements for the equipment of the AL controlled by a Proto program such as Blotter. Similar setup functions could be programmed for a Proto program operating an entirely different piece of automated equipment, such as a machine tool. Also, the system can be programmed so that by selecting Group Samples from the Setup menu, the system displays the dialog window of Fig. 19. This is an example of a window that is used to represent physical features of the apparatus. For example, the 8 by 12 array of positions in Fig. 19, identified by numbered columns and letters for rows, can be used to represent a 96 position vial tray that is used in the AL for samples. A user can change the groupings in the graphic array and also the labels for the groupings by the text fields. Clicking on OK can be used to save changes. The designations thus provided tell the system where to look for sample groups when called by icons in a process flow.

Similarly, the system can be programmed such that selecting Restriction Groups in the preferred embodiment displays the dialog window of Fig. 20. The text fields in the window are for an operator to use to identify certain groups of chemicals and enzymes to be used in a process sequence and to be available for loading from various positions in the AL. Four of the fields in the dialog window of Fig. 20 have scroll bars at one side. These fields are known as scrolling fields, and one can use the arrows and scroll bars to scroll through lists too long to display in the field in their entirety.

Similarly, the system can be programmed so that selecting Select Probes in the preferred embodiment displays the dialog window of Fig. 21. The Select Probes dialog window is for a user to identify probes relative to sample groups.

Appendix A provides an index to Proto with a brief explanation of each subroutine. Appendix B provides a listing of each of the implemented subroutines. Fig. 22 shows a diagram illustrating the Proto code structure.

It will be clear to one with skill in the art that although the many functions in the preferred embodiment have been described with reference to the Blotter process, many different chemical processes could be programmed for the AL in a similar way. It will be clear, too, that similar functions would be useful for automatic control of various other kinds of processes amenable to control by Proto. For example, in applying Proto to an NC milling machine, one could provide functions for relating certain tool bits with different tool holders and for plotting speeds and feeds for operation.

It will be apparent to one with skill in the art that there are many changes that may be made without departing from the spirit and scope of the invention. A software application program according to the invention may be programmed for any of a large number of different computers, so the invention is not restricted to the Apple Macintosh machines that will run the preferred embodiment described herein. Proto programs may also be prepared for a very large number of different kinds of process equipment, not just for the AL machine described herein as being operated by Proto. There are many, many kinds of icons that can be drawn, and the icons may or

may not have text associated with them. There are similarly many ways one might implement and initiate specific functions in Proto, and still maintain the essentials of the invention. For example, the pull-down menus in the menu bar are not essential, but merely convenient. Pop Up menus that appear in response to keyboard commands, and other types, could also be used. There are a large variety of similar kinds of changes that could be made without departing from the spirit and scope of the invention.

What is claimed is:

1. A computer system for generating and sending electronic signals to processing equipment to initiate and sequence actions performed by said processing equipment, said computer system comprising:

computer means for processing input signals and generating output signals;

display terminal means connected to said computer means for providing a visual display to a user;

input means for accepting said input signals from a user and conveying said input signals to said computer means; and

software means for directing the process performance of said computer means and the interaction of said computer means with said user;

said software means providing an interactive display on, and in conjunction with, said display terminal means, said interactive display comprising:

at least one top level graphic icon representing a step-by-step process, said top level icon expandable in place in said display in response to a user-initiated signal to show first lower level icons in said step-by-step process, said top level icon by expanding in place becoming an identified boundary surrounding said first lower level icons, maintaining the visual relationship between the top level process and the first lower level icons, said first lower level icons connected by lines representing a sequence of performance

in process flow, said first lower level icons each expandable in place to reveal next lower level connected icons until a fundamental icon is displayed, said fundamental icon being expandable in place to display an interactive input control panel for accepting data specific to a fundamental process step.

2. A computer system as in claim 1 wherein said input means comprises keyboard means and mouse means.

3. A computer system as in claim 1 wherein hardware equipment is Macintosh equipment made and sold by Apple Computer Company.

4. A computer system as in claim 1 wherein said expansion in place is activated by a user by double clicking on an icon with a mouse.

5. A computer system as in claim 1 wherein each said expanded boundary comprises interactive signal means for initiating collapsing of said boundary and all display within said boundary back to said related icon.

6. A computer system as in claim 1 wherein said interactive display in any state of expansion or collapse comprises a single screen, said screen pannable by a user to show the entire process represented by said interactive display.

7. A computer system as in claim 1 wherein said interactive display comprises functions whereby a user may insert and remove icons in the process flow, thereby altering the process flow signalled by said computer to said processing equipment.

8. A computer system as in claim 1 wherein at least one of said first lower level icons is expandable in place to reveal next lower level connected icons until a fundamental icon is displayed, said fundamental icon being expandable in place to display an interactive input control panel for accepting data specific to a fundamental process step.

9. A method for programming an automated apparatus to perform a process, comprising:

arranging a sequence of first icons on a display in the order of said process, wherein said first icons represent functions of said apparatus, wherein at least one of said first icons provides a visual representation of a function of said apparatus;

wherein said at least one of said first icons can be expanded to show second icons that comprise the function of said at least one of said first icons, at least one of said second icons being a visual representation of a subfunction of the apparatus.

10. The method of claim 9 wherein, when said at least one of said first icons is expanded, said at least one of said first icons maintains its same sequential relationship on said display to the other of said first icons in said sequence as before it was expanded.

AMENDED CLAIMS

[received by the International Bureau on 25 March 1991 (25.03.91); original claims 1-10 replaced by amended claims 1-10 (3 pages)]

1. A computer system for generating and sending electronic signals to processing equipment to initiate and sequence actions performed by said processing equipment, said computer system comprising:

computer means for processing input signals and generating output signals;

display terminal means connected to said computer means for providing a visual display to a user;

input means for accepting said input signals from a user and conveying said input signals to said computer means; and

software means for directing the process performance of said computer means and the interaction of said computer means with said user;

said software means providing an interactive display on, and in conjunction with, said display terminal means, said interactive display comprising:

at least one top level graphic icon representing a step-by-step process, said top level icon expandable in place in said display in response to a user-initiated signal to show first lower level icons in said step-by-step process, said top level icon by expanding in place becoming an identified boundary surrounding said first lower level icons, maintaining the visual relationship between the top level icon and the first lower level icons, said first lower level icons connected in a manner representing a sequence of performance in process flow.

2. A computer system as in claim 1 wherein at least one of said first lower level icons is expandable in place to reveal next lower level connected icons until a fundamental icon is displayed, said fundamental icon being expandable in place to display an interactive input control panel for accepting data specific to a fundamental process step.

3. A computer system as in claim 1 wherein said input means comprises keyboard means and mouse means.

4. A computer system as in claim 1 wherein said expansion in place is activated by operation of said mouse means.

5. A computer system as in claim 1 wherein each said expanded boundary comprises interactive signal means for initiating collapsing of said boundary and all display within said boundary back to said related icon.

6. A computer system as in claim 1 wherein said interactive display in any state of expansion or collapse comprises a single screen, said screen pannable by a user to show the entire process represented by said interactive display.

7. A computer system as in claim 1 wherein said interactive display comprises functions whereby a user may insert and remove icons in the process flow, thereby altering the process flow signalled by said computer to said processing equipment.

8. A computer system as in claim 1 further comprising means for selecting graphic icons from a memory system, displaying said icons on said display terminal means, and arranging said icons in a connected sequence representing an iconic computer program which causes the processing equipment to carry out the sequence of functions represented by the sequence of icons..

9. A method for programming an automated apparatus to perform a process, comprising:

arranging a sequence of first icons on a display in the order of said process, wherein said first icons represent functions of said apparatus, wherein at least one of said first icons provides a visual representation of a function of said apparatus;

wherein said at least one of said first icons can be expanded to show second icons that comprise the function of said at least one of said first icons, at least one of said second icons being a visual representation of a subfunction of the apparatus.

10. The method of claim 9 wherein, when said at least one of said first icons is expanded, said at least one of said first icons maintains its same sequential relationship on said display to the other of said first icons in said sequence as before it was expanded.

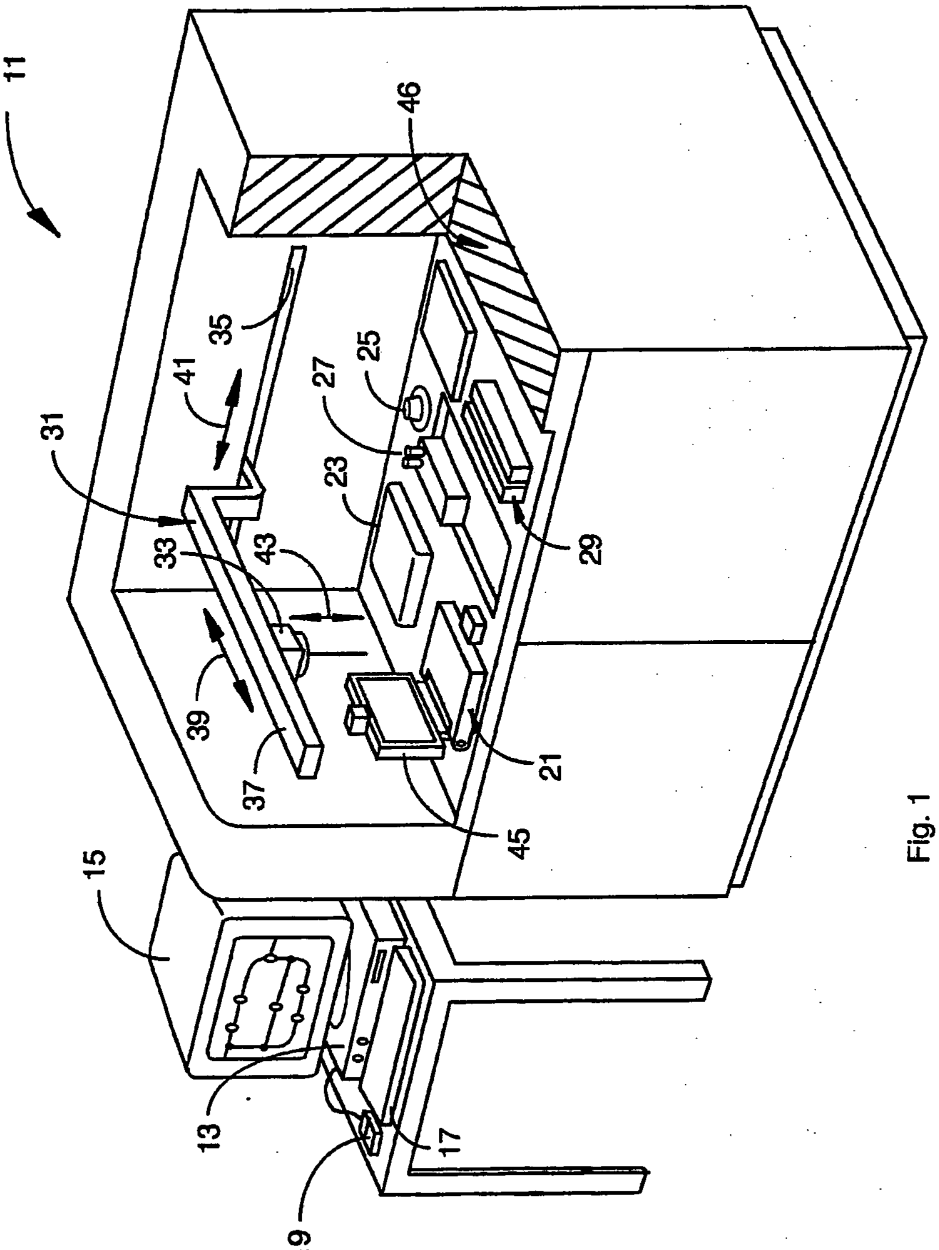


Fig. 1

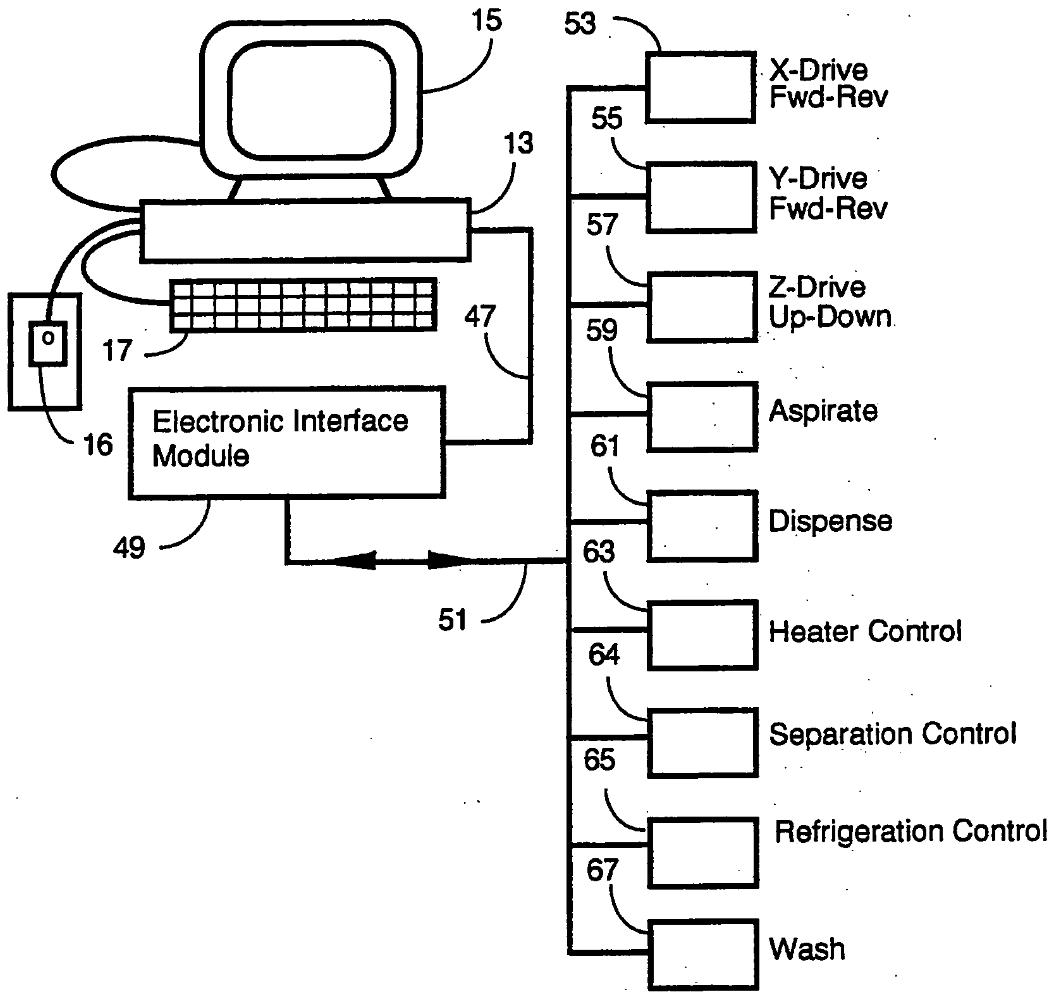

15
53
X-Drive
Fwd-Rev
55
13
Y-Drive
Fwd-Rev
57
Z-Drive
Up-Down
47
17
59
Aspirate
Electronic Interface
Module
16
61
Dispense
49
63
51
Heater Control
64
Separation Control
65
Refrigeration Control
67
Wash

Fig. 2

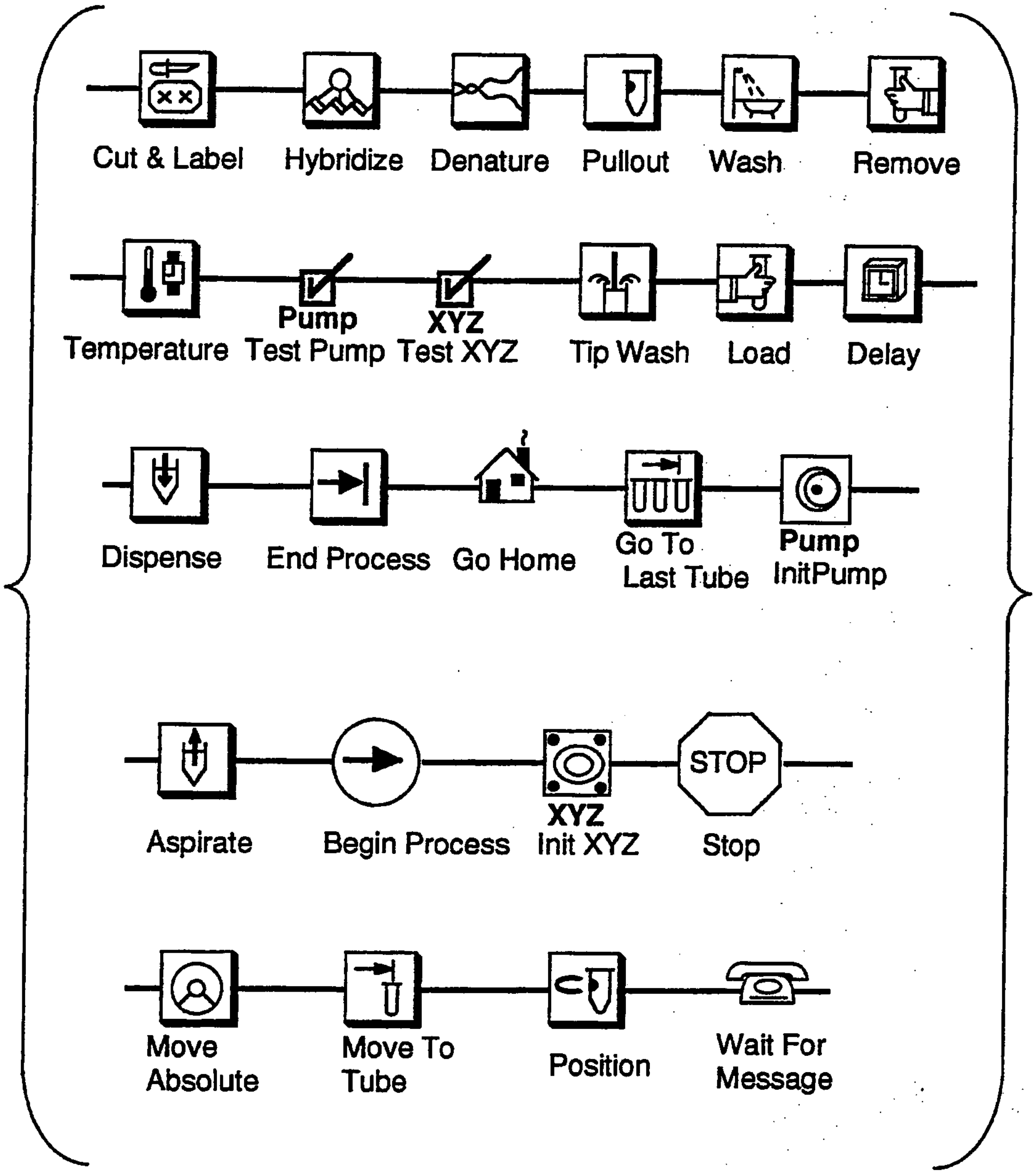

Cut & Label
Hybridize
Denature
Pullout
Wash
Remove
Temperature
Pump
Test Pump
XYZ
Test XYZ
Tip Wash
Load
Delay
Dispense
End Process
Go Home
Go To Last Tube
Pump
InitPump
Aspirate
Begin Process
XYZ
Init XYZ
STOP
Stop
Move Absolute
Move To Tube
Position
Wait For Message

Fig. 3

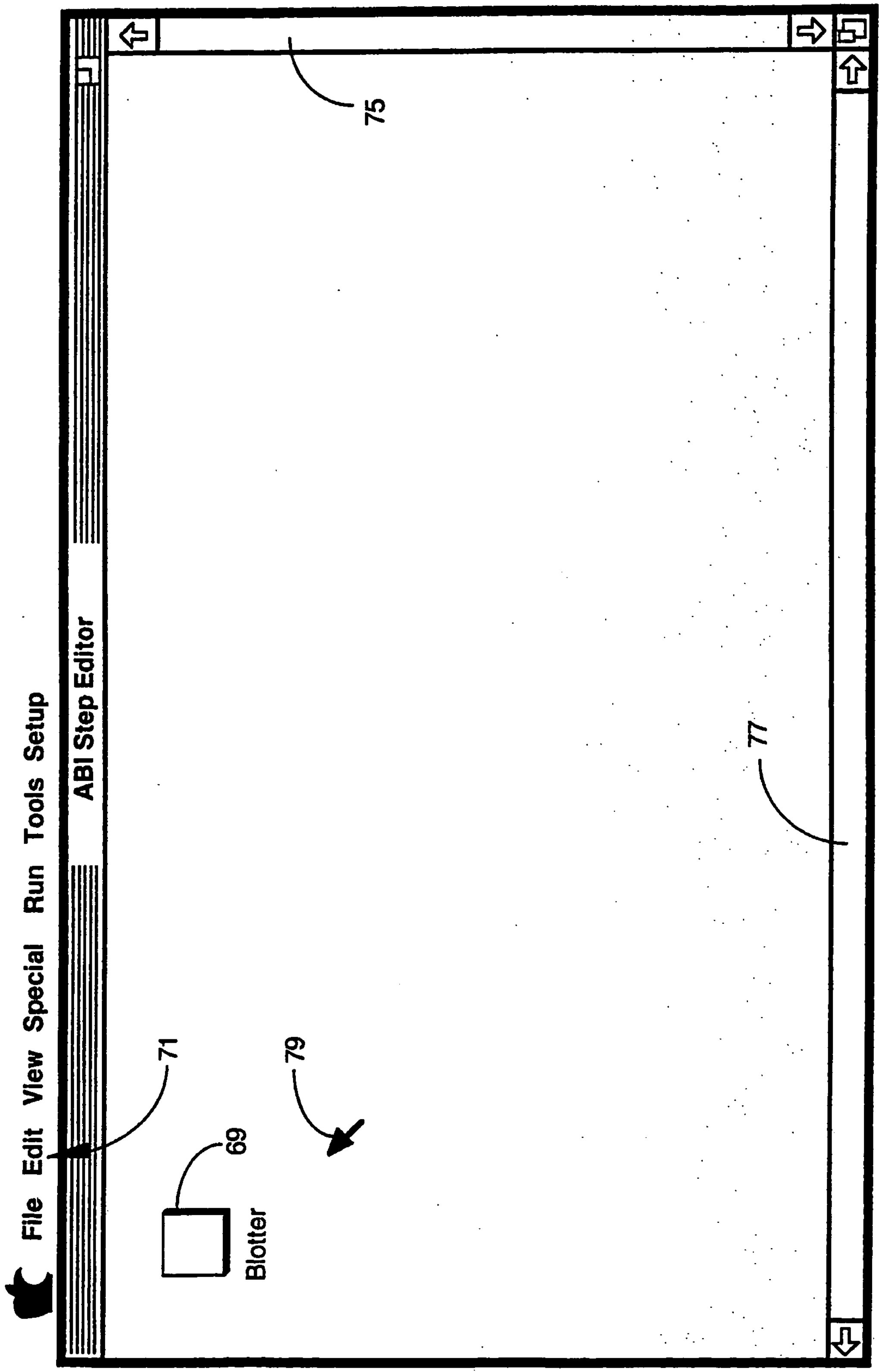

File Edit View Special Run Tools Setup
ABI Step Editor
Blotter
69
71
75
77
79

Fig. 4

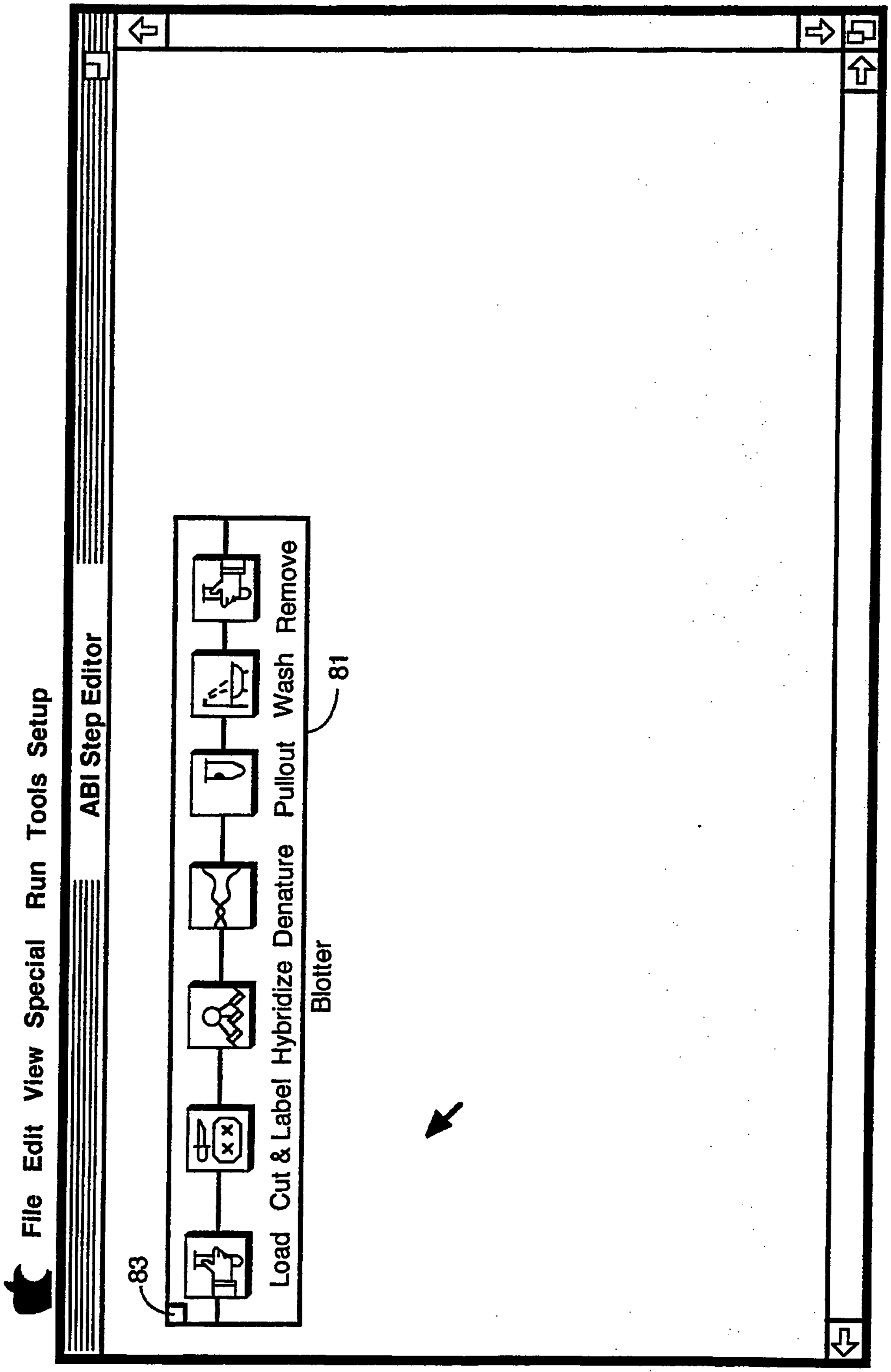

File Edit View Special Run Tools Setup
ABI Step Editor
83
Load
Cut & Label
Hybridize
Denature
Pullout
Wash
Remove
Blotter
81

Fig. 5

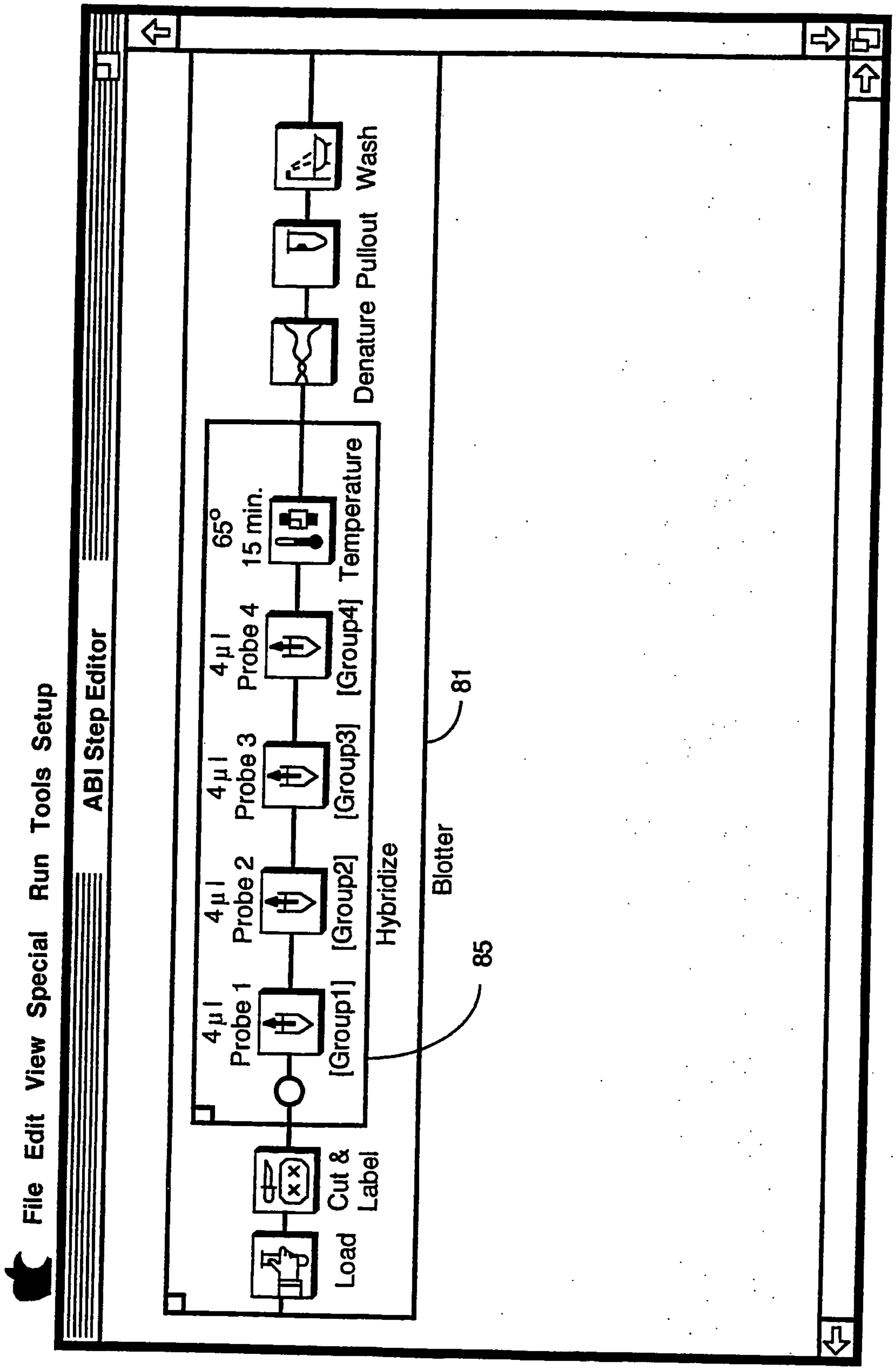

File Edit View Special Run Tools Setup
ABI Step Editor
Load
Cut & Label
4 µl Probe 1 [Group1]
4 µl Probe 2 [Group2]
4 µl Probe 3 [Group3]
4 µl Probe 4 [Group4]
65° 15 min. Temperature
Hybridize
Denature
Pullout
Wash
Blotter
81
85

Fig. 6

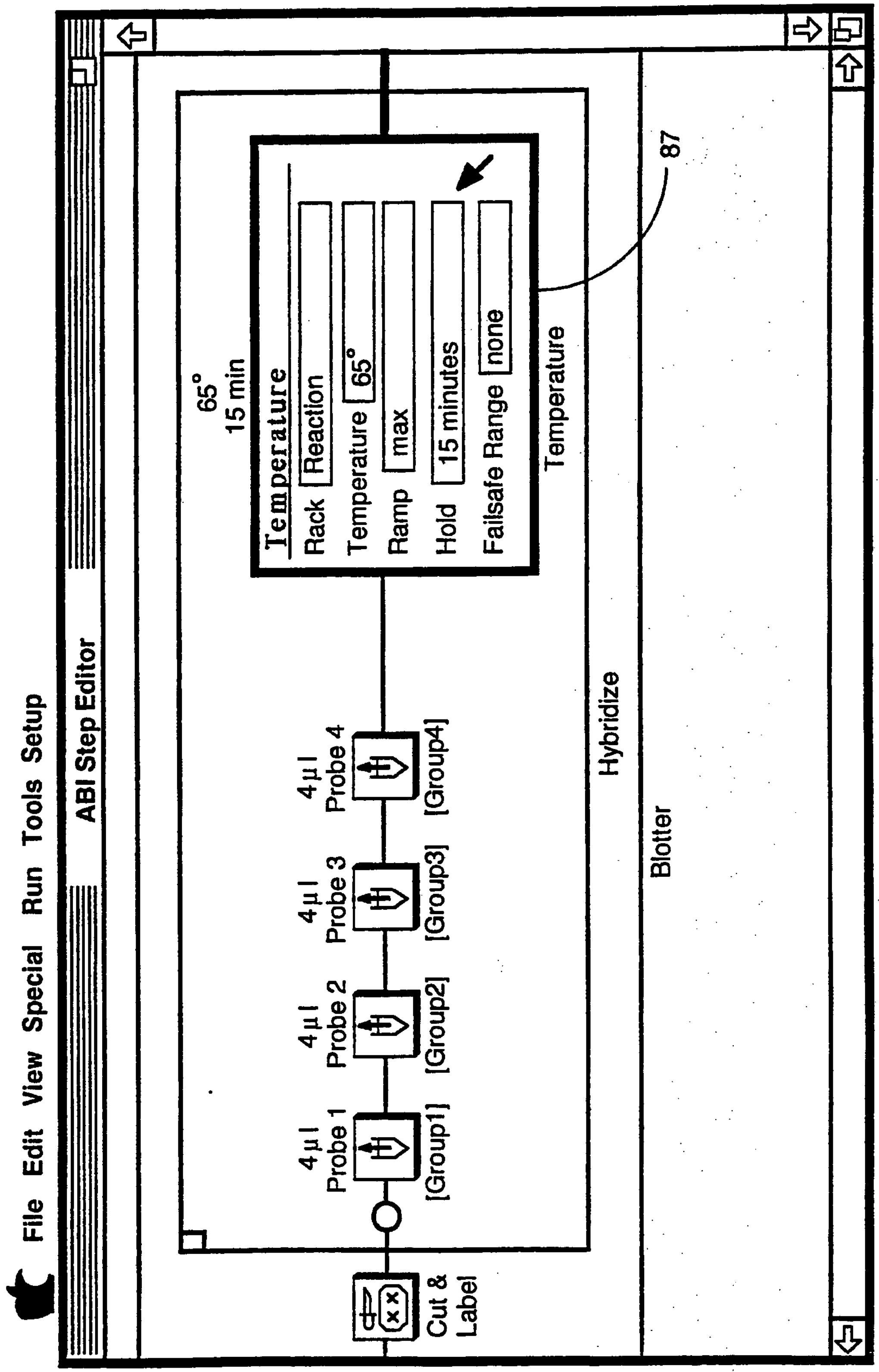

File Edit View Special Run Tools Setup
ABI Step Editor
Cut & Label
4 µl Probe 1 [Group1]
4 µl Probe 2 [Group2]
4 µl Probe 3 [Group3]
4 µl Probe 4 [Group4]
65° 15 min
Temperature
Rack Reaction
Temperature 65°
Ramp max
Hold 15 minutes
Failsafe Range none
Temperature
Hybridize
Blotter
87

Fig. 7

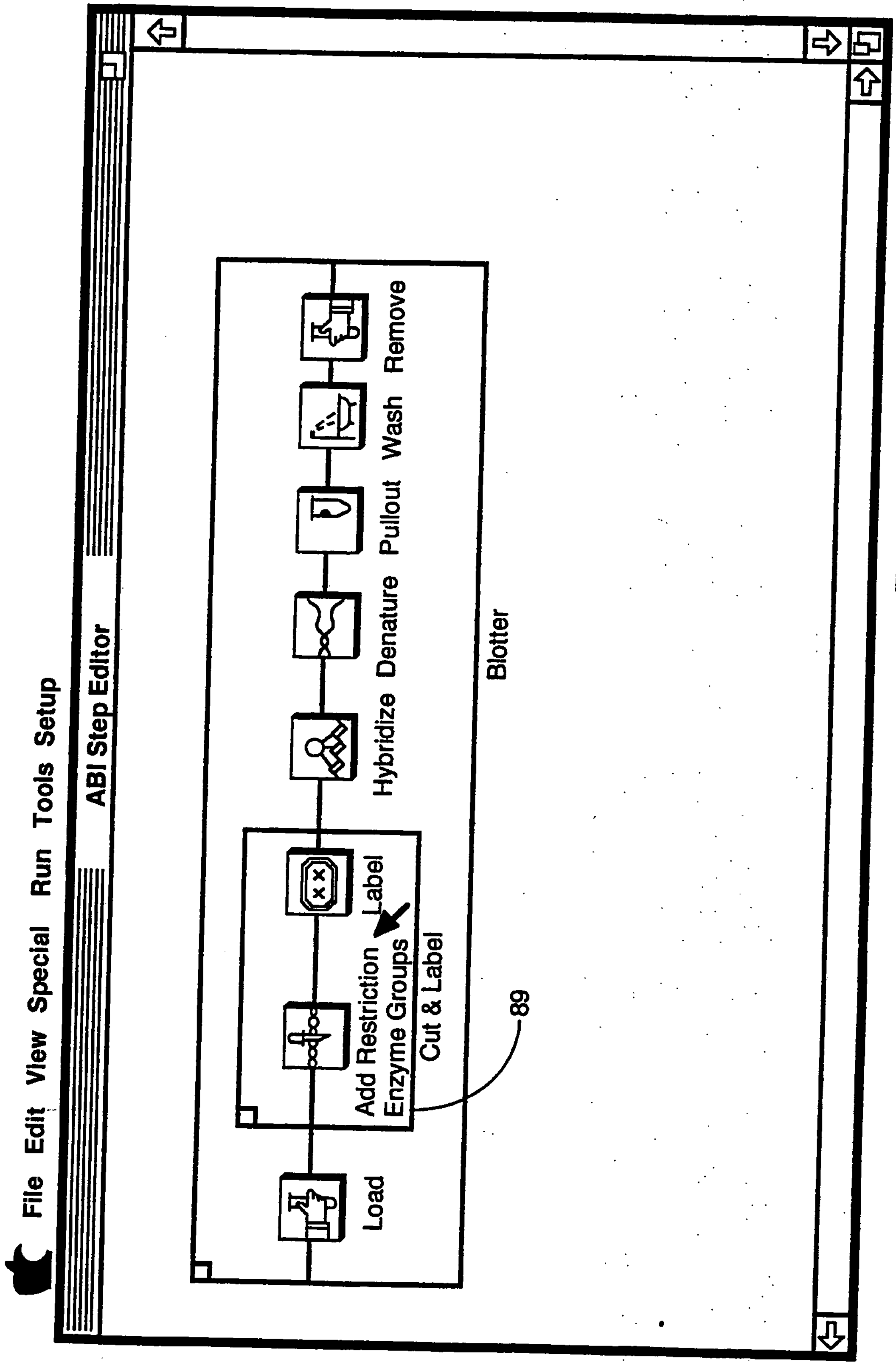

File Edit View Special Run Tools Setup
ABI Step Editor
Load
Add Restriction Enzyme Groups
Label
Cut & Label
89
Hybridize
Denature
Pullout
Wash
Remove
Blotter

Fig. 8

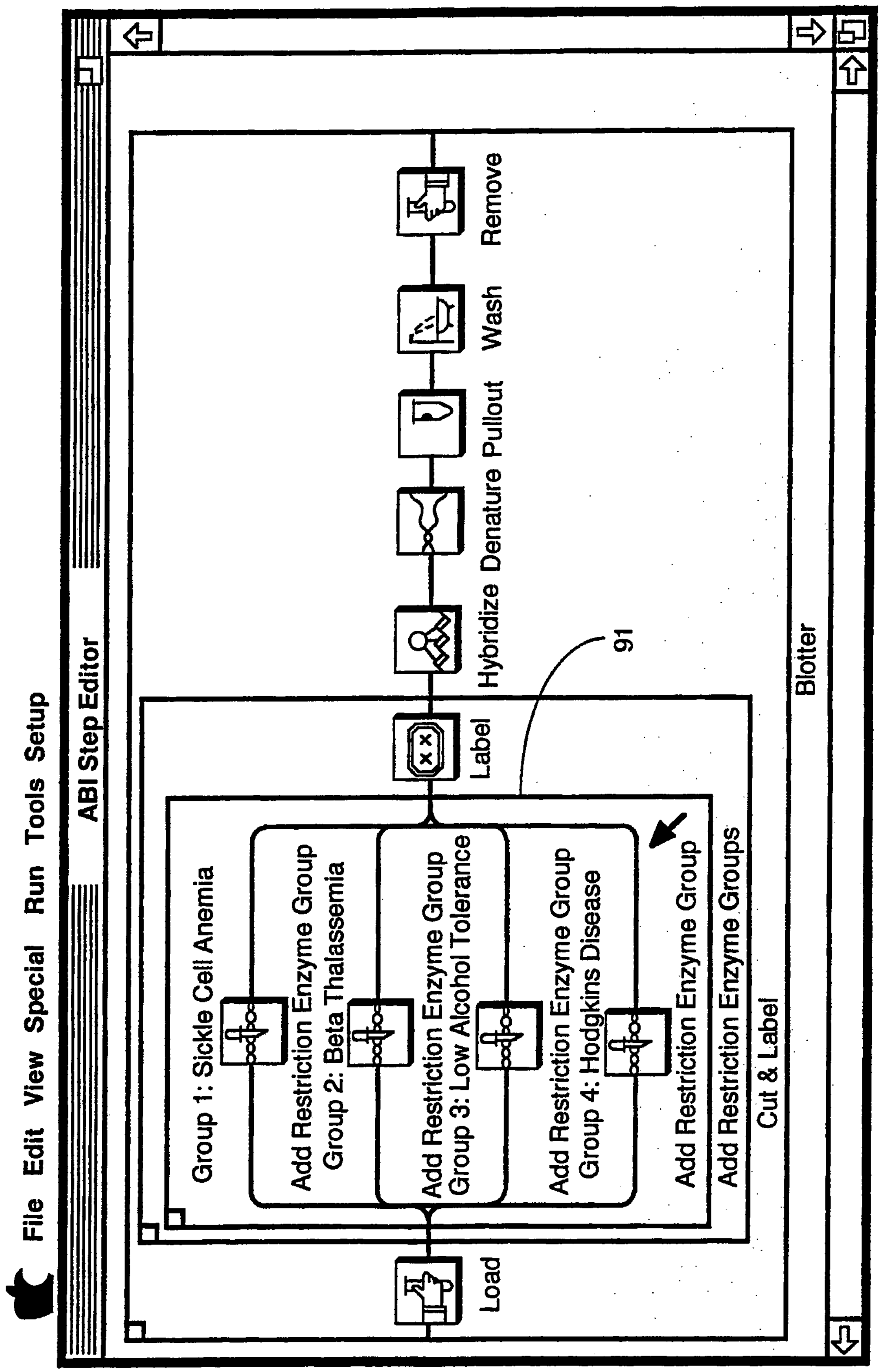

File Edit View Special Run Tools Setup
ABI Step Editor
Load
Group 1: Sickle Cell Anemia
Add Restriction Enzyme Group
Group 2: Beta Thalassemia
Add Restriction Enzyme Group
Group 3: Low Alcohol Tolerance
Add Restriction Enzyme Group
Group 4: Hodgkins Disease
Add Restriction Enzyme Group
Add Restriction Enzyme Groups
Cut & Label
Label
91
Hybridize
Denature
Pullout
Wash
Remove
Blotter

Fig. 9

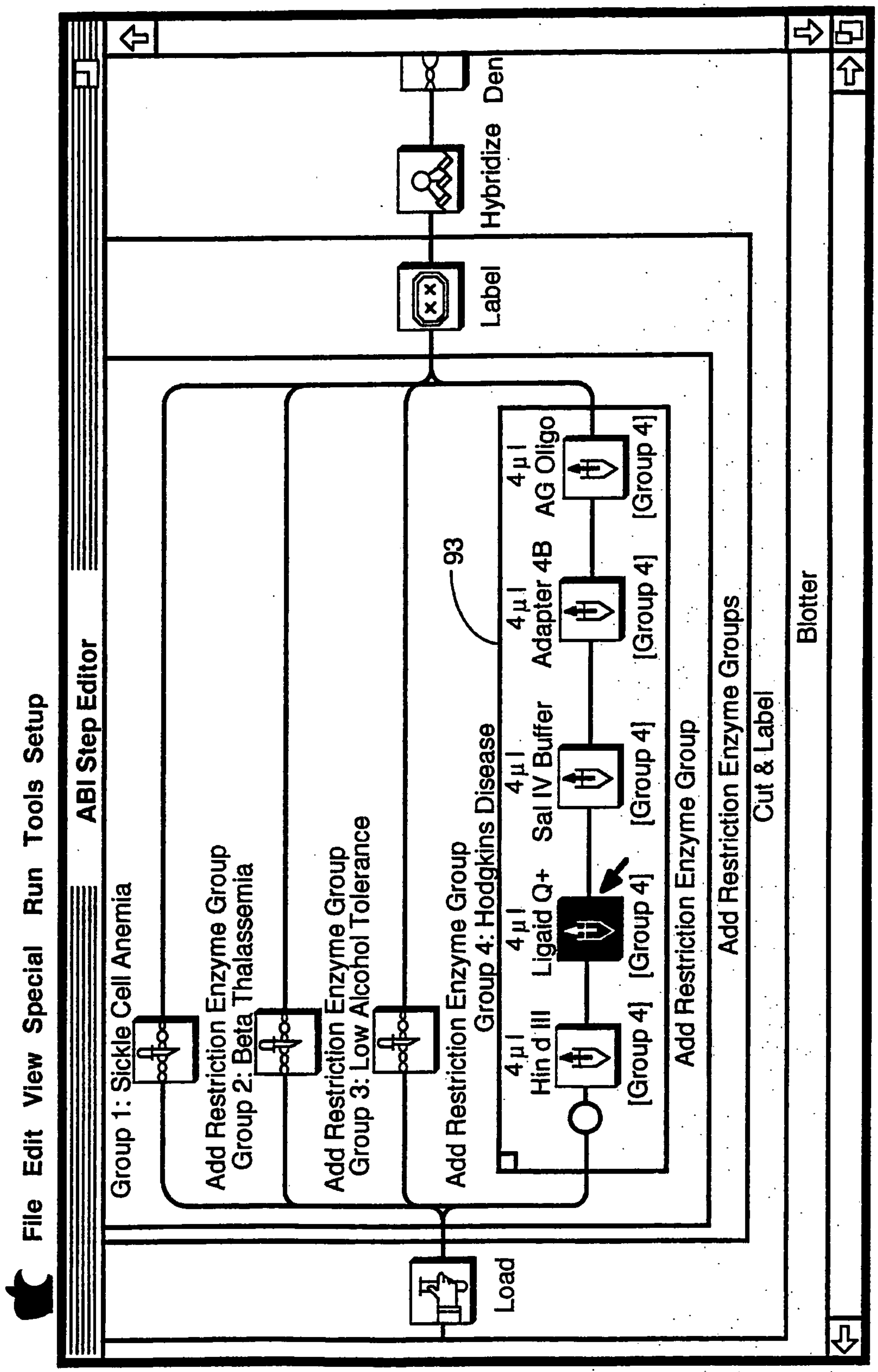

File Edit View Special Run Tools Setup
ABI Step Editor
Group 1: Sickle Cell Anemia
Add Restriction Enzyme Group
Group 2: Beta Thalassemia
Add Restriction Enzyme Group
Group 3: Low Alcohol Tolerance
Add Restriction Enzyme Group
Group 4: Hodgkins Disease
93
4 µl
Hin d III
4 µl
Ligaid Q+
4 µl
Sal IV Buffer
4 µl
Adapter 4B
4 µl
AG Oligo
[Group 4]
[Group 4]
[Group 4]
[Group 4]
[Group 4]
Add Restriction Enzyme Group
Add Restriction Enzyme Groups
Cut & Label
Blotter
Load
Label
Hybridize
Den

Fig. 10

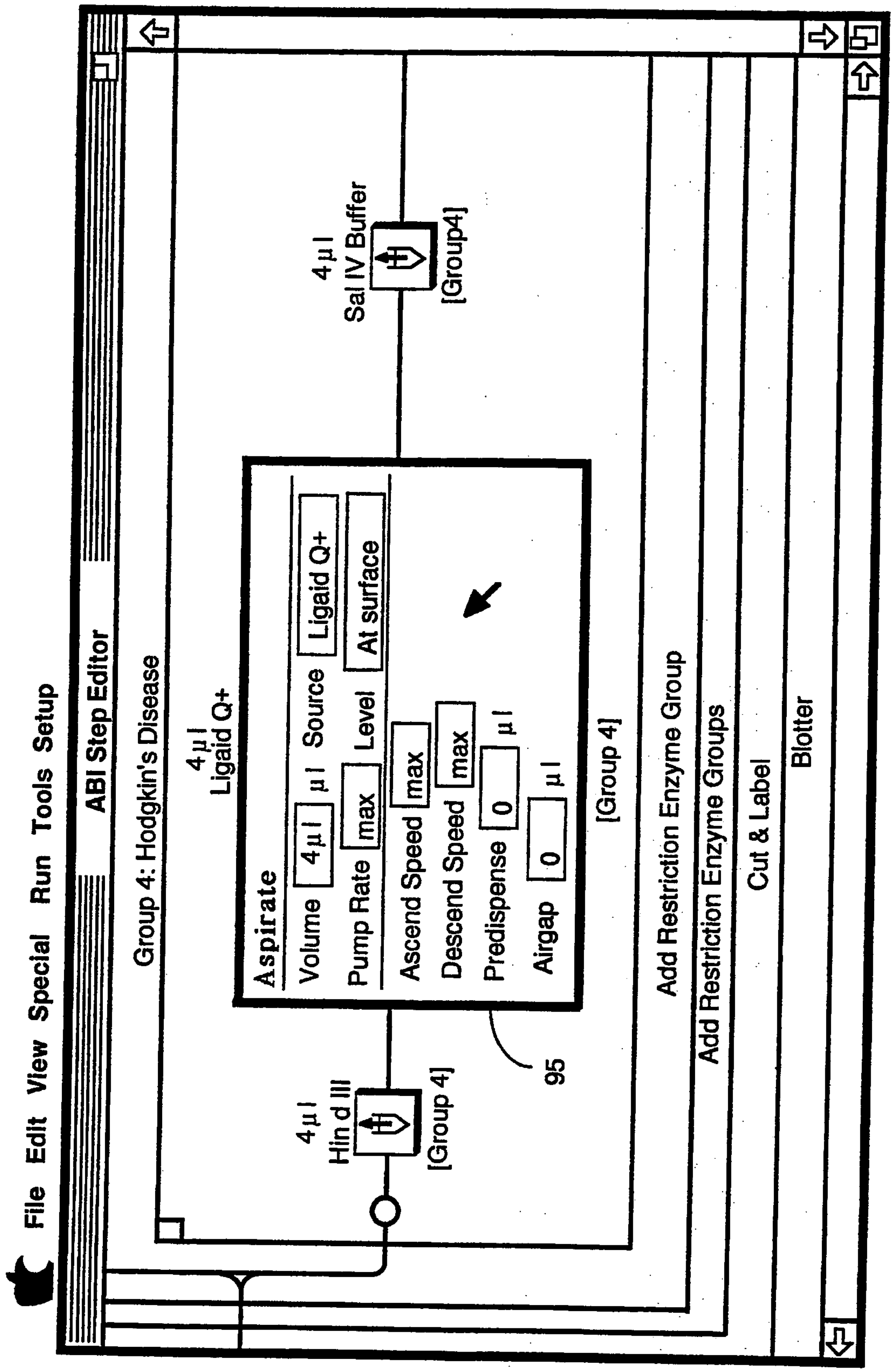

File Edit View Special Run Tools Setup
ABI Step Editor
Group 4: Hodgkin's Disease
4 µl
Ligaid Q+
Aspirate
Volume
4 µl
µl
Source
Ligaid Q+
Pump Rate
max
Level
At surface
Ascend Speed
max
Descend Speed
max
Predispense
0
µl
Airgap
0
µl
[Group 4]
4 µl
Hin d III
[Group 4]
95
4 µl
Sal IV Buffer
[Group4]
Add Restriction Enzyme Group
Add Restriction Enzyme Groups
Cut & Label
Blotter

Fig. 11

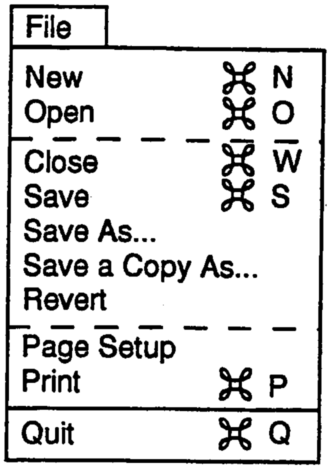


Fig. 12

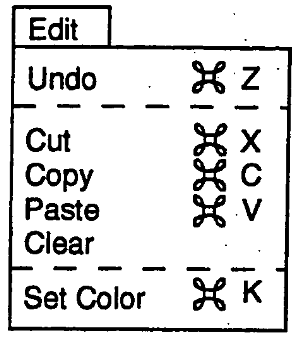


Fig. 13

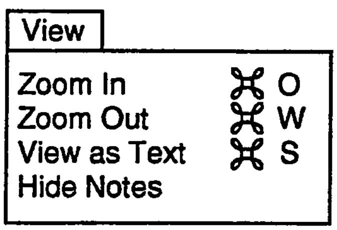


Fig. 14

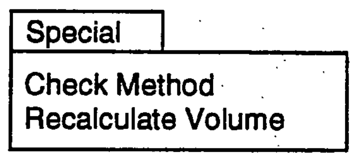


Fig. 15

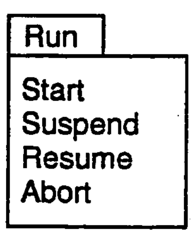


Fig. 16

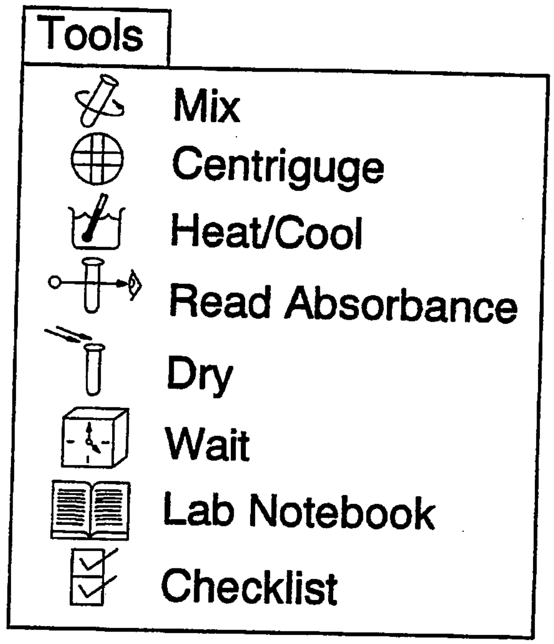

Tools
Mix
Centriguge
Heat/Cool
Read Absorbance
Dry
Wait
Lab Notebook
Checklist

Fig. 17

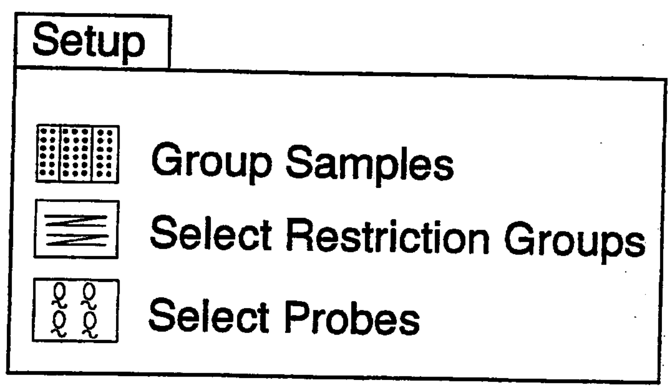

Setup
Group Samples
Select Restriction Groups
Select Probes

Fig. 18

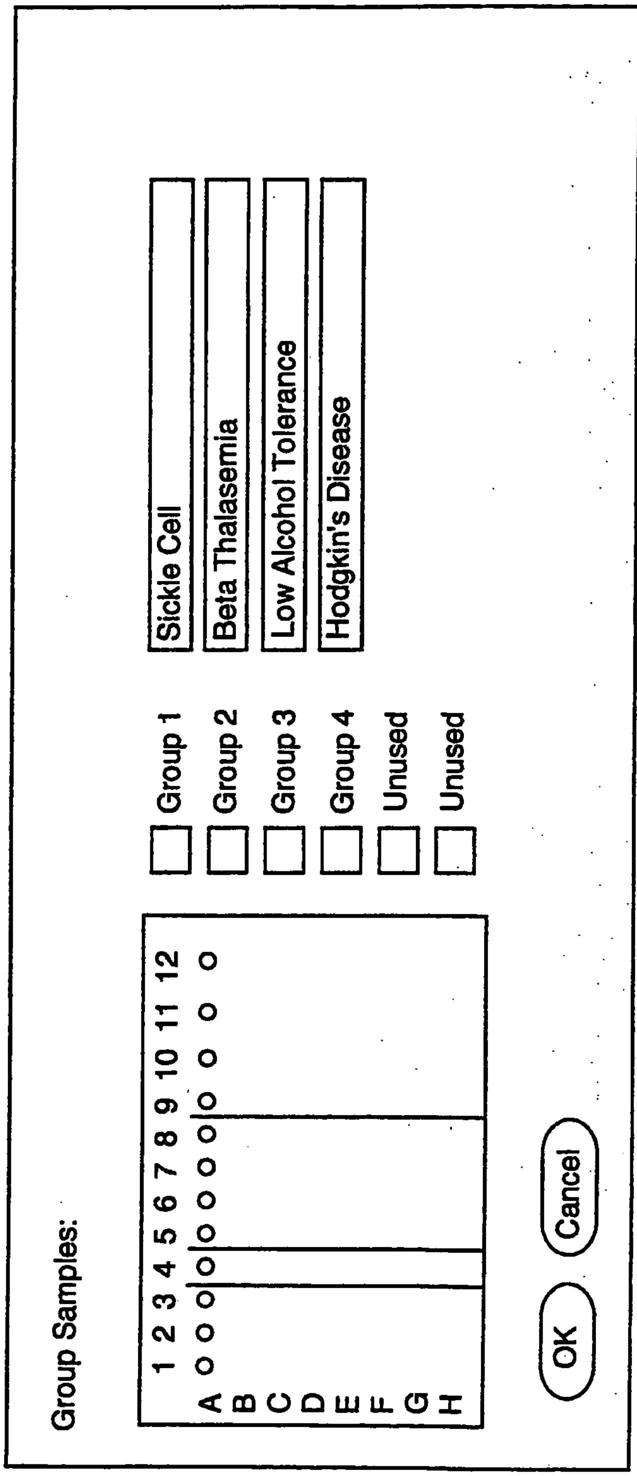

Group Samples:
1 2 3 4 5 6 7 8 9 10 11 12
A B C D E F G H
Group 1
Group 2
Group 3
Group 4
Unused
Unused
Sickle Cell
Beta Thalasemia
Low Alcohol Tolerance
Hodgkin's Disease
OK
Cancel

Fig. 19

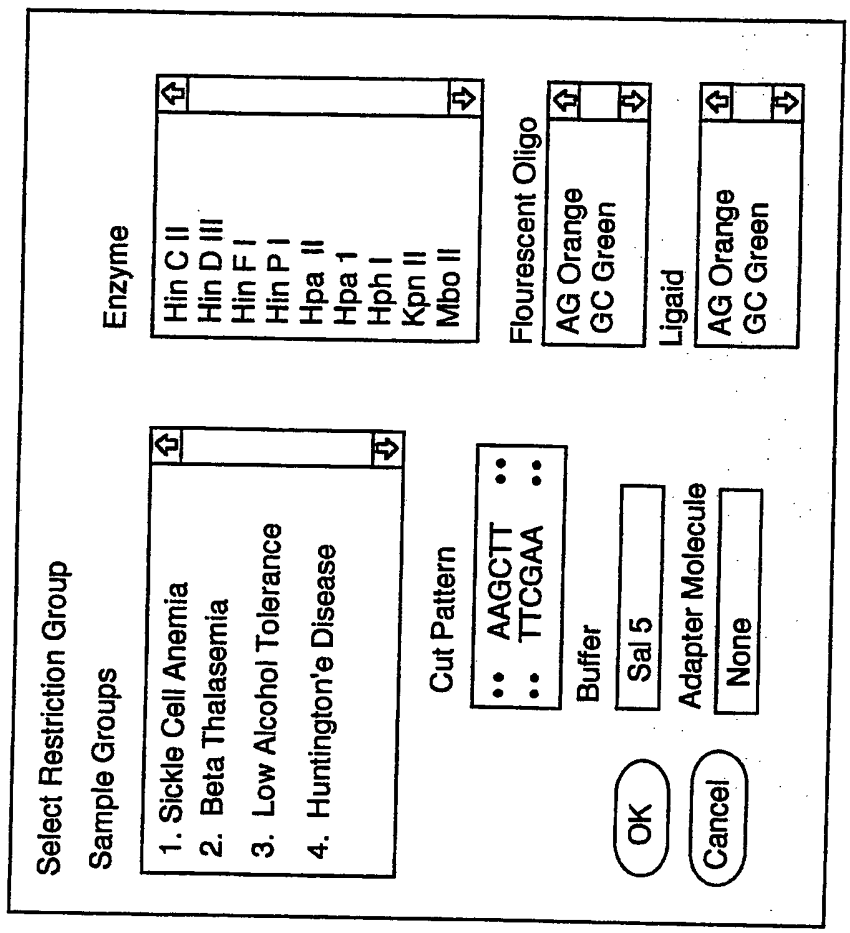

Select Restriction Group
Sample Groups
1. Sickle Cell Anemia
2. Beta Thalasemia
3. Low Alcohol Tolerance
4. Huntington'e Disease
Enzyme
Hin C II
Hin D III
Hin F I
Hin P I
Hpa II
Hpa 1
Hph I
Kpn II
Mbo II
Cut Pattern
•• AAGCTT ••
•• TTCGAA ••
Flourescent Oligo
AG Orange
GC Green
Buffer
Sal 5
Ligaid
AG Orange
GC Green
Adapter Molecule
None
OK
Cancel

Fig. 20

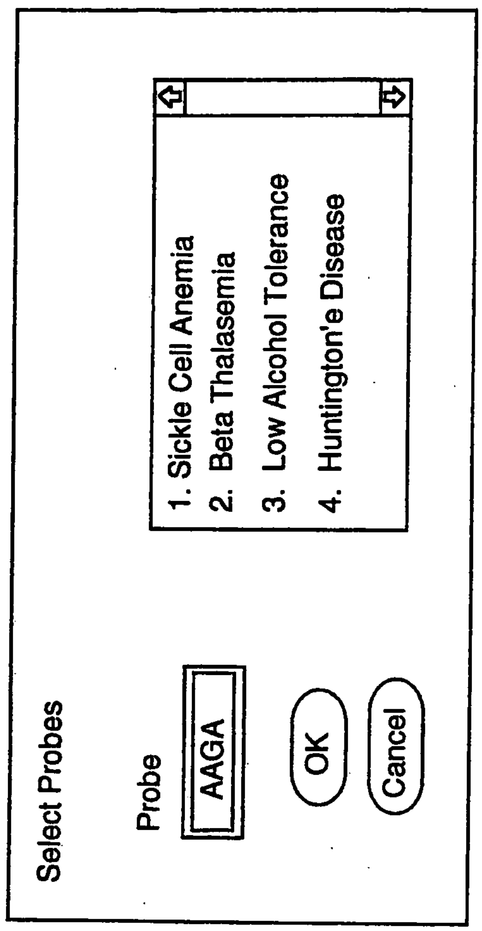

Select Probes
Probe
AAGA
OK
Cancel
1. Sickle Cell Anemia
2. Beta Thalasemia
3. Low Alcohol Tolerance
4. Huntington'e Disease

Fig. 21

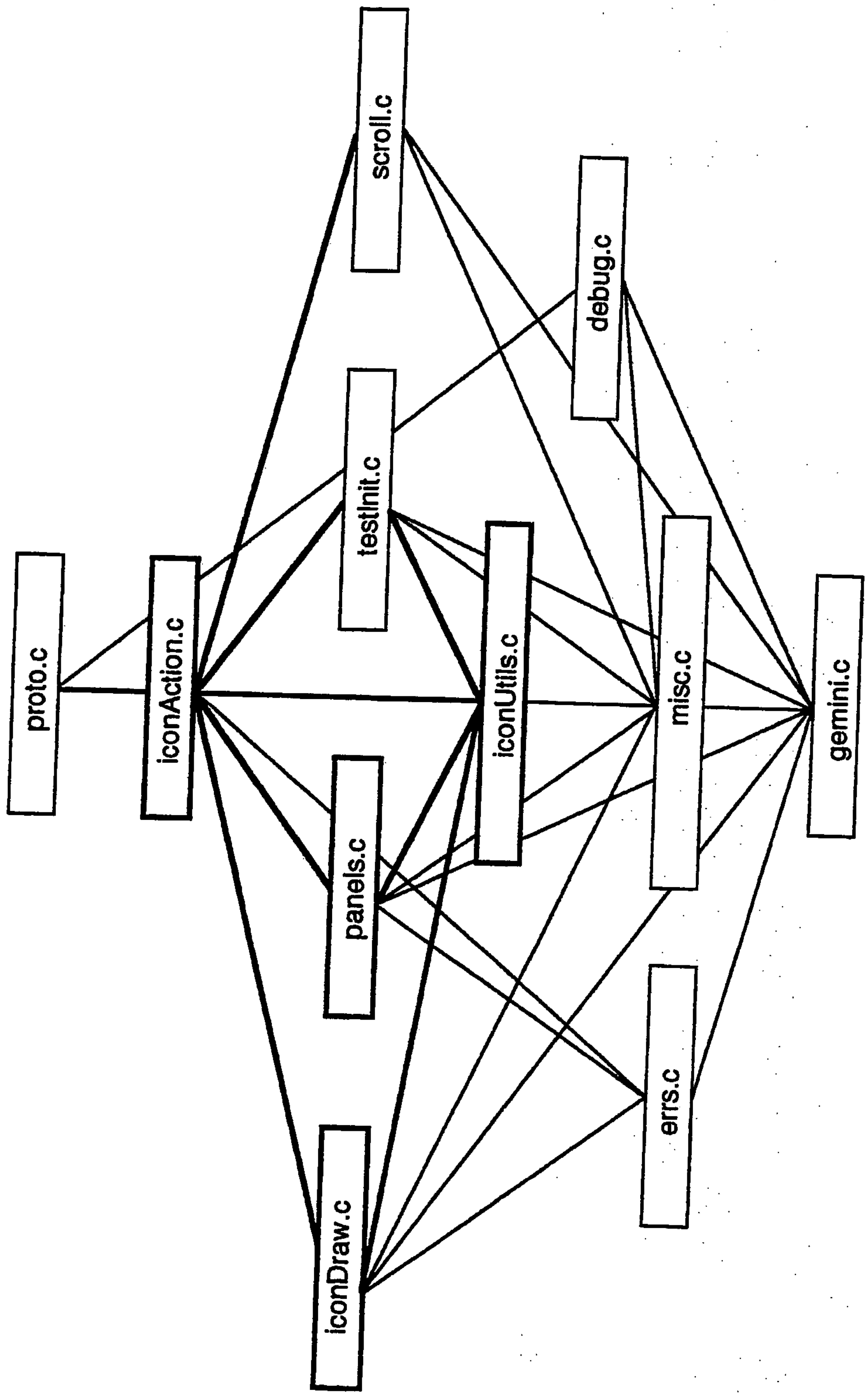

scroll.c
debug.c
testInit.c
proto.c
iconAction.c
iconUtils.c
misc.c
gemini.c
panels.c
errs.c
iconDraw.c

Fig. 22

INTERNATIONAL SEARCH REPORT

International Application No PCT/US90/06000

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [3]

According to International Patent Classification (IPC) or to both National Classification and IPC
INT. CL (5): G06F 3/14
U.S. CL: 364/521

II. FIELDS SEARCHED

Minimum Documentation Searched [4]

Classification System	Classification Symbols
U.S.	364/518,521,522,513 340/725,747

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [5]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [14]

Category *	Citation of Document, [16] with indication, where appropriate, of the relevant passages [17]	Relevant to Claim No. [18]
A,P	US, A, 4,959,799 (YOSHIURA ET AL.) 25 September 1990, see figure 9.	1-10
X	US, A, 4,549,275 (SUKONICK) 22 October 1985 see figures 1A and 1B.	1-10
X,E	US, A, 4,965,743 (MALIN ET AL.) 23 October 1990 see figure 2.	1-10
X	US, A, 4,860,204 (GENDRAN ET AL.) 22 August 1989 see figure 6.	1-10
A	US, A, 4,613,946 (FARMAN) 23 September 1986 see figure 3.	1-10
A	US, A, 4,642,780 (THOMSON) 10 February 1987 see figures 2-3.	1-10

* Special categories of cited documents: [15]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search [2]	Date of Mailing of this International Search Report [2]
31 December 1990	12 FEB 1991
International Searching Authority [1]	Signature of Authorized Officer [20]
ISA/US	Ngoc-Ho Nguyen NGUYEN NGOC-HO PHU K. NGUYEN INTERNATIONAL DIVISION

Form PCT/ISA/210 (second sheet) (May 1986)

THIS PAGE BLANK (USPTO)

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ____________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.